Hindawi
Computational Intelligence and Neuroscience
Volume 2021, Article ID 5590445, 15 pages
https://doi.org/10.1155/2021/5590445

Research Article

Learning Intuitive Physics and One-Shot Imitation Using State-Action-Prediction Self-Organizing Maps

Martin Stetter[1] **and Elmar W. Lang**[2]

[1]*Department of Bioengineering Sciences, Weihenstephan-Triesdorf University of Applied Sciences, Freising D-85354, Germany*
[2]*Computational Intelligence and Machine Learning Group, Department of Biophysics, University of Regensburg, Regensburg D-93053, Germany*

Correspondence should be addressed to Martin Stetter; martin.stetter@hswt.de

Received 18 January 2021; Revised 14 October 2021; Accepted 21 October 2021; Published 12 November 2021

Academic Editor: Jianli Liu

Human learning and intelligence work differently from the supervised pattern recognition approach adopted in most deep learning architectures. Humans seem to learn rich representations by exploration and imitation, build causal models of the world, and use both to flexibly solve new tasks. We suggest a simple but effective unsupervised model which develops such characteristics. The agent learns to represent the dynamical physical properties of its environment by intrinsically motivated exploration and performs inference on this representation to reach goals. For this, a set of self-organizing maps which represent state-action pairs is combined with a causal model for sequence prediction. The proposed system is evaluated in the *cartpole* environment. After an initial phase of playful exploration, the agent can execute kinematic simulations of the environment's future and use those for action planning. We demonstrate its performance on a set of several related, but different one-shot imitation tasks, which the agent flexibly solves in an active inference style.

1. Introduction

During the last decade, rapid progress in the field of deep learning has led to a number of remarkable achievements in many fields of artificial intelligence (AI) [1]. However, human learning and intelligence seem to work radically differently from the supervised pattern recognition approach adopted in most deep learning architectures. Among many other things, humans, for example, playing infants, are able to learn from exploration and imitation, learn from much fewer examples, and create richer representations [2]. They can flexibly reason over these representations and creatively elicit novel state configurations never seen before.

Here we suggest a simple neural network architecture which learns to represent the dynamic physical characteristics of its environment in an unsupervised, exploratory way. By inference on the basis of this representation, the system can plan actions to reach externally given or intrinsically generated goals. In the following, we summarize related work and outline the proposed model.

Impressive performance of deep learning approaches has been demonstrated in some classical supervised tasks such as object [3] or speech recognition [4, 5], but also in unsupervised domains including representation learning and learning generative models [6–8]. Recently, "deep reinforcement learning" [9] has been demonstrated to achieve human- or superhuman-level performance on playing Atari video games from raw pixel frames.

Despite of all these achievements, studying deep learning might be less useful when it comes to understanding humanlike intelligence with the goal to create artificial general intelligence [2]. Major issues raised include the following:

(i) Deep learning approaches are in essence model-free and require massive amounts of labelled examples—orders of magnitude more than humans—in order to learn a complex task [2, 10].

(ii) In most deep learning approaches, the original problem is being reformulated in a clever way as a related, supervised task which can be tackled by

deep multilayer perceptrons. Hence, the intelligent, creative part is done by the human designer, not by the algorithm.

(iii) Deep learning approaches work on associations and cannot grasp cause and effect [11].

The first issue has been addressed in many ways with approaches summarized as "few-shot learning" (for a recent review, see [12]). Generally, a model is trained on a large body of related tasks to learn an inductive bias or prior, which is then exploited to solve the task at hand based on only one or few examples. Impressive recent achievements include one-shot imitation learning in robots [13] and meta-reinforcement learning [14], the latter showing close relationships to biological reinforcement learning [15].

Yet, the systems trained are extremely complex in terms of the amount of parameters, and consequently the amount of data required, when summing over the different tasks, is still very high.

Addressing the second and third issues seems to require a qualitative change in the paradigm of an artificial intelligent system, compared to deep learning.

Recently, Lake et al. [2] formulated cornerstones believed to be crucial ingredients of humanlike learning and cognition, which is suggested to be more model-building-like than pattern-recognition-like: (i) "building causal models" of the world, (ii) "ground models on intuitive theories of physics and psychology", and (iii) "harness compositionality and learning-to-learn." The authors state that in the approach of learning as model-building:

"Cognition is about using these models to understand the world, to explain what we see, to imagine what could have happened and did not, or what could be true and is not, and then planning actions to make it so." ([2], p. 2).

Lake et al. [16] have developed "probabilistic program induction" as a model for few-shot visual concept learning, which focusses on compositionality and learning-to-learn.

In our approach, which we outline in the next but one section, we address the first issue by deliberately designing a very simple (i.e., strongly constrained yet biologically plausible) model. Despite being simple, the model allows addressing the second and third issues by being able to actively explore the causal structure of its environment, to learn intuitive physics, and to plan actions in order to achieve goals.

In this work, we suggest a very simple neural architecture, which learns in completely unsupervised fashion and incorporates several of the mentioned principles: it learns a model of the dynamics of its environment by playful exploration ("intuitive physics"), can play virtual, predicted episodes ("what could be true and is not"), and can plan action sequences to bring the environment closer to a target state that has been given extrinsically by a one-shot demonstration ("planning actions to make it so"). Conceptually, the same mechanism could be utilised, if the agent could generate intrinsic goals.

The proposed system consists of sparse unsupervised networks, which in this work are implemented as Kohonen-type self-organizing maps (SOM) [17]: a perceptual or *state* network, which learns a representation of the environment's states, an *action* module which represents possible motor commands, and a sensorimotor integrating *state-action* network which learns and represents associations between both (Figure 1(a)). During "playful exploration," the agent executes (e.g., randomly sampled) actions and observes resulting state changes. Observing means that each active state-action unit learns to predict the environment's next state given the currently active state-action pair it represents. For this, a transition model is learned by updating state transition matrices on top of the SOMs. This mechanism discovers the effects caused by the agent's own actions. For convenience, we refer to this architecture as "state-action-prediction self-organizing maps" (SapSom). When proposing this architecture, the authors are well aware that mere temporal order does not necessarily reflect a true cause-effect relationship: the rooster crows before sunrise, but it does not cause sunrise. Nevertheless, the rooster's crow can be used to predict sunrise with a decent hit-rate.

In the proposed setup, playing virtual episodes or "kinematic mental simulation," for which evidence has been found in human reasoning [18], corresponds to repeated prediction of state transitions starting from a virtual start state under a virtual action sequence. Here, the term "virtual" denotes intrinsically generated states, which are represented by active units without sensory stimulation and without action to be executed. This process of playing virtual episodes serves to transform a virtual action sequence into a predicted sequence of states.

A goal is defined as a target state or group of states in latent state space. Action planning corresponds to searching in action sequence space in the style of active inference, such that predicted resulting states approximate as good as possible a desired region in latent space (i.e., reach that goal).

The active inference paradigm adopted here [19–21] represents the action generation mechanism proposed in the predictive processing (PP) paradigm of brain modelling [22–24]. PP models view the cerebral cortex as a hierarchically organized prediction engine which constantly tries to explain (i.e., predict) incoming sensory data as effect of hidden causes. Successful prediction is suggested to cause the subjective percept. Hence, sensor signals act as supervisory signals, and the brain's internal states (hypothesized causes) play the role of generative signals or inputs, which are adjusted such as to minimize prediction error. Active inference refers to the process of acting on the environment such as to make the own prediction come true.

The paper is organized as follows. An introductory section on related work is followed by the Model section, where we specify the model architecture including the representation and prediction learning procedures and the mechanism of inference and action planning. In the Method and Results section thereafter, we provide a proof of concept using the Open AI's gym cartpole environment. The section is subdivided into three parts: First, SapSom is trained by playfully exploring the cartpole environment using random action sequences. It is shown that the agent correctly learns to represent the phase space structure of the cartpole system, referred to as "intuitive physics." In the second part, the

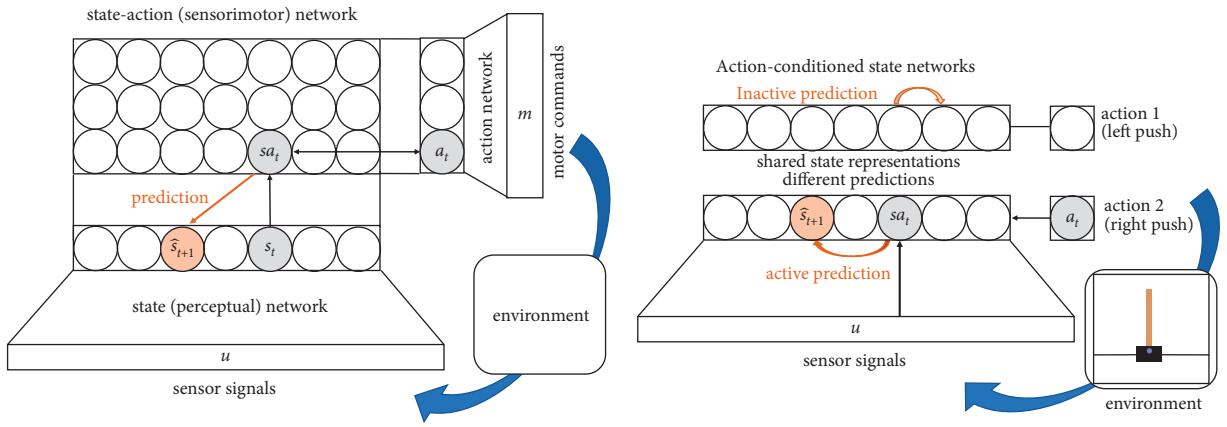

Figure 1: (a) Proposed network architecture. A sensorimotor self-organizing map learns to represent state-action combinations, each represented by a state and action SOM, respectively. An activated state-action unit learns to predict the most likely next state, $\hat{s}_{t+1}$ (brown), conditioned on the current state and action, s_t, a_t it represents. (b) Reduced architecture actually implemented for the demonstrations in the result section (for details, see text). 1D state and action representations are drawn for simplicity.

ability to perform kinematic simulation is examined by comparing real and simulated dynamics of the environment under three prespecified deterministic action sequences as well as under many random action sequences. A quantitative error measure is provided which demonstrates that predicted state sequences closely resemble the actual temporal state sequences provided by the environment. In the third part of the Method and Results section, we analyze the ability of the agent to reach various goals, which are extrinsically given as successful example state sequences. Three classes of goals are considered—the balancing task, the controlled-tilt task, and the tilted-balancing task—and individual runs for visual inspection as well as a quantitative performance assessment are given. It is shown that the agent flexibly and most of the time successfully solves many instances of these tasks after observing only one demonstration. We conclude the paper by discussing how our model might be extended to yield more powerful architectures and relating our model to biological findings. A preprint of this work has been made available under [25].

2. Related Work

In contrast to current trends in deep learning, human like learning systems need to build causal models of their environments (causality), acquire an intuitive understanding of the underlying physics (intuitive physics), and develop learning-to-learn skills and combining constituents (compositionality) to generalize knowledge to new tasks and events [2]. As such, learning intuitive physics has become a major target of recent research activities in cognitive sciences and artificial intelligence. Humans seem to have a basic understanding of objects and their physical interactions from infancy [26], which becomes refined during development by self-curated experiments with their environments. This curiosity-driven learning of intuitive physics [27] helps humans to mentally run physics simulations [28] in order to plan and predict future actions and events. Furthermore, a task-to-task transfer and generalization to unseen events remains a big challenge to any artificial intelligence system.

The SapSom model presented below emphasizes simplicity in the sense of strongly constrained and therefore easy-to-train architecture. A set of SOMs is equipped with a simple yet powerful additive, Hebb-trained prediction system. In spirit, it shows close relationship to model-predictive control. There are similar approaches which address intuitive physics learning on the basis of self-organizing maps. For example, Toussaint [29] designed a sensorimotor SOM, in which lateral connections of a sensory map are modified by motor activations in a multiplicative, modulatory way. Morasso et al. [30] address the targeting movement problem in robotics by setting up one-joint SOM for sensory and motor spaces. In contrast to these approaches, we use several SOMs with action-conditioned transition models, which in consequence can become much simpler to train than in the mentioned approaches. Apart from this class of approaches with limited complexity, a large number of intuitive physics learning and planning systems have been proposed, which show relationships to deep learning.

A common approach to learn intuitive physics is via handcrafted simulation engines [28] or via physics engines made adaptable through deep neural networks [31, 32]. More generally, dynamics of object interactions have been learned either jointly with perception [33] or through decomposition of visual scenes into structured representations of objects and their dynamic states [34, 35]. Alternatively, perception modules have been further developed to be able to either imagine future states of objects [36] or to predict them through a GAN-based approach or an encoder-decoder network [37]. With a similar goal, a graph-based perception-prediction-network (PPN) has been proposed in [38] to learn without human intervention latent object properties from their interactions. During a gradient-based training with samples of object dynamics, the perception module generates representations of object properties, while

a prediction module uses the latter to simulate system dynamics. The PPN seamlessly generalizes to unseen scenarios, and its learned representations can be translated into human-interpretable properties. As discussed in [39], structured representations through graph networks provide a strong relational inductive bias to support reasoning and combinatorial generalization. The perception-prediction concept has also been harnessed by Wu et al. [40] to learn intuitive physics. There, a perception module first learns a representation of the physical environment, which is then used by physics and graphics engines to learn interpreting and reconstructing the visual stimulus sequences. Later, the generative models are used for reasoning and prediction. Inverting a physics engine is achieved by a convolutional inversion network.

As sensor signals are often fraught with uncertainty, models of the outside world need to integrate sensory inputs with internal prior beliefs, in accordance with Bayesian inference. Hence, probabilistic approaches are able to deal with uncertainty and, combined with system identification techniques like particle filters, achieve robust long-term predictions of object dynamics while simultaneously learning the underlying intuitive physics [41, 42].

Several recent approaches refer to learning physics from observations alone. Applying exploratory learning of causal relationships was the intention of Kansky et al. [43] when designing an object-oriented generative physics simulator. It is based on a complex structured network architecture and can learn the dynamics of an environment only from observations. By reasoning backward through causes, this Schema Network can transfer intuitively learned physics to unseen situations. An unsupervised learning of intuitive physics purely from visual observations was reported in [44]. Unsupervised predictors of physical states were constructed by tracking salient objects in dynamic stimulus sequences based on causality and equivariance and using the learned physical states to train visual predictors, which successfully could incorporate the underlying environment. With a similar goal, Ehrhardt et al. [45] implemented an unsupervised meta-learning formulation to predict the dynamics of moving objects in a complex environment and to generalize to a varying number of moving objects. The network architecture comprises sequential autoencoders as well as a U-net encoder-decoder to extract the relevant information. Finally, an image sequence is predicted employing a stack fully convolutional network, while the parameters of the system are adapted by optimizing a proper loss function. Meta-learning was also used to train neural networks augmented with an autoencoder-based meta-recognition model producing model code, which a subsequent meta-generative model uses to construct parameters for task-specific models. This meta-learning autoencoder (MeLA) framework is able to build models for previously unseen tasks, which closely match the true underlying models [46]. A simulator-augmented interaction networks (SAIN) model was proposed in [47], which combines object-based networks, which learn residuals, with an object-based learnable physics engine to search for actions in control problems.

Rather than relying on physics engines, Nguyen et al. [48] design a surprise and explain (SnE) framework, which rests upon the basic premise that object dynamics is mostly linear, and any nonlinear dynamics presents a surprise for which an explanation cannot be inferred easily. The anomaly detection framework comprises three modules: perception, dynamics, and explanation. An interpretable intuitive physics model has also been designed by Ye et al. [49]. The bottleneck layers of the deep network architecture contain specific dimensions, which correspond to different physical properties. The model is trained with sequences of colliding objects and generalizes well to scenarios with different underlying physical properties.

3. Model

In this work, we actually implement a reduced version of the model outlined in Figure 1(a). The resulting simplified architecture is shown in Figure 1(b). Simplifications are as follows: (i) no action map is explicitly learned, which would maintain codebook vectors of motor commands. This restricts the implementation to discrete finite action spaces, which is, however, sufficient for the cases studied here. (ii) Instead of a full sensorimotor map, a set of $\mathbf{K}$ action-conditioned state-only maps is trained, where $\mathbf{K}$ is the number of actions. All maps share the same perceptual (state) representation, but on top of each map an individual, action-conditioned state transition matrix is learned. This prohibits dimensional control of the sensorimotor space but makes sure that all state-action pairs are readily represented.

3.1. Representation Learning. We wish to learn a sparse representation of the input space, because it is believed that this will facilitate the learning of unimodal sharply peaked state transition distributions. As discussed above, many powerful techniques for representation learning and even several possibilities for obtaining sparse representations are around (e.g., [50]). Here we use self-organizing maps (SOMs) [17], because in addition to sparsity they maintain topographic order in map space, which may be useful in terms of predictive processing.

SOMs have been successfully used both in bioinspired hierarchical representation learning [51] and reinforcement learning [52]. To briefly summarize, in a SOM, neurons are geometrically arranged in a regular grid (here 2D rectangular). Each unit at map location $\mathbf{s} \in \mathbb{R}^2$ maintains a codebook vector $\mathbf{w}(\mathbf{s})$ with the same dimension as the input space. On presentation of an input vector or "environmental state" $\mathbf{u}$, the winning unit $\mathbf{s}^*$ is found, defined by its codebook vector being closest to the input

$$\mathbf{s}^*(\mathbf{u}) = \operatorname{argmin}_{\mathbf{s}} \|\mathbf{u} - \mathbf{w}(\mathbf{s})\|, \tag{1}$$

where $\|\cdot\|$ denotes Euclidean norm, and the notation on the left-hand side of (1) explicates the dependence of the winner on $\mathbf{u}$. The winning unit and its neighbours in map space learn according to

$$\Delta \mathbf{w}(\mathbf{s}) = \eta h_{\mathbf{s}^*(\mathbf{u})}(\mathbf{s})(\mathbf{u} - \mathbf{w}(\mathbf{s})),$$
$$h_{\mathbf{s}^*(\mathbf{u})}(\mathbf{s}) = \exp\left(-\frac{\|\mathbf{s}^*(\mathbf{u}) - \mathbf{s}\|^2}{2\sigma^2}\right), \quad (2)$$

where h can be interpreted as a localized neural activation pattern centered around the winning unit and σ is defined in (3). The strong localization of activation entails that only a small fraction of units is active at every time step, resulting in a sparse code. The learning rule (2) brings codebook vectors of map neighbours both closer to regions of high data density and to each other.

3.1.1. SOMs and Predictive Processing. There is an interpretation of SOMs in terms of predictive processing: when considering the codebook vector of an active unit $\mathbf{s}$ as prediction of the input, $\|\mathbf{u} - \mathbf{w}(\mathbf{s})\|$ is just the prediction error in input space (the term "prediction" being used in the sense of "predicting the presence"), often referred to as quantization error. Finding $\mathbf{s}^*(\mathbf{u})$ according to (1) thus minimizes the input prediction error by inference, and the learning step (2) further minimizes it by learning.

Similar to [51], we define a data-driven variable width σ of the activation pattern; namely,

$$\sigma = \sigma_0 \frac{\min_{\mathbf{s}} \|\mathbf{u} - \mathbf{w}(\mathbf{s})\|}{\text{mean}_{\mathbf{s}} \|\mathbf{u} - \mathbf{w}(\mathbf{s})\|}, \quad (3)$$

where σ_0 has been empirically set to 25 percent of the map diameter throughout this work. By this, for inputs that are well-represented with small prediction error, only a small neighbourhood learns, whereas a large prediction error (maybe due to nonstationary statistics of the input) causes a large portion of the map to rearrange to better represent this novel input.

The normalized activation pattern in map space can then be interpreted as recognition density, where

$$p(\mathbf{s}|\mathbf{u}) = \frac{h_{\mathbf{s}^*(\mathbf{u})}(\mathbf{s})}{\sum_{\mathbf{s}'} h_{\mathbf{s}^*(\mathbf{u})}(\mathbf{s}')}. \quad (4)$$

According to (3), the recognition density is sharply peaked for small prediction errors and more distributed for larger representational uncertainty. In SOMs, the generative distribution, which denotes the likelihood of inputs given a map state, simply becomes $p(\mathbf{u}|\mathbf{s}) = \delta(\mathbf{u} - \mathbf{w}(\mathbf{s}))$, δ denoting Kronecker's delta. For a comprehensive treatment of generative and recognition models, see, for example, [19–21].

Finally, due to topographic order, prediction error in map space can be defined in geometrical terms, simply as geometric distance between most likely true and predicted states, respectively.

3.2. Prediction Learning. During learning, the system will exploit the possibility to act on the environment and to directly observe the consequences of these own actions on environmental state changes. Hence, learning is situated at the *intervention* level of Pearl's Causal Hierarchy and can be formulated in the framework of *Do*-calculus [53]. Due to its similarity to how infants explore their environment by testing the effects of their actions, we metaphorically refer to this style of learning as "playful exploration."

During playful exploration, the model learns to approximate the action-conditioned Markov transition distribution $p(\mathbf{s}'|\mathbf{s}, do(a))$, where $\mathbf{s}', \mathbf{s}$ run over all state units and a runs over all actions. The distribution denotes the probability of finding map state $\mathbf{s}'$ activated one time step after in state $\mathbf{s}$ action a has been executed. For this, each action-conditioned state network, labelled by a, updates an individual state transition matrix $\mathbf{T}_a$ with components $T_a(\mathbf{s}', \mathbf{s})$ (for matrix operations, indices $\mathbf{s}', \mathbf{s}$ being appropriately reordered as scalars), whenever a is executed. Let $\mathbf{p}_t$ be the column vector of probabilities $p(\mathbf{s}|\mathbf{u}_t)$, (4), assigned to map states at time t in response to input $\mathbf{u}_t$. On execution of action a_t, the environment changes to state $\mathbf{u}_{t+1}$ leading to a new distribution $\mathbf{p}_{t+1}$. The state distribution *predicted* by network a_t, in contrast, is given by $\widehat{\mathbf{p}}_{t+1} = \mathbf{T}_{a_t} \cdot \mathbf{p}_t$. We adopt a simple least squares scheme and minimize $\|\mathbf{p}_{t+1} - \mathbf{T}_{a_t} \cdot \mathbf{p}_t\|^2$ with respect to $\mathbf{T}_{a_t}$. Gradient descent leads to the learning rule

$$\Delta \mathbf{T}_{a_t} = \gamma(\mathbf{p}_{t+1} - \widehat{\mathbf{p}}_{t+1})\mathbf{p}_t^T, \quad (5)$$

where γ is a learning step size variable and the superscript $(\cdot)^T$ denotes the transpose of a vector or matrix. We prefer least squares over minimization of the Kullback–Leibler divergence between both distributions, which is often pursued, because the latter is usually tractable only under severe simplifications, which often lead to treatment of modes only.

3.3. Inference. After exploring the environment to a sufficient extent, inference can be done on the so-far learned representation using the state transition matrices. For example, given a certain start map state $\mathbf{s}_0$, the system can generate virtual action sequences by activating action nodes due to some schedule, without actually executing the corresponding motor commands, and predict the sequence of states that would result from executing that sequence. Correlates of this in human cognition might be kinematic mental simulation with the goal of planning actions. Moreover, the start state might be virtually generated as well, instead of perceptually caused, giving the possibility to elaborate on virtual scenarios never seen before, which might be considered a kind of artificial creativity. We do not formulate an explicit neuronal model of how state and action representations might be spontaneously generated, but there are mechanisms and models of how this might occur spontaneously or in response to stimulation [54]. In the present context, sequence prediction is referred to as "playing virtual episodes," as this procedure accepts a start state and an action sequence and returns a sequence of predicted environmental states.

3.3.1. Sequence Prediction. One-step prediction given environmental state $\mathbf{u}_t$ and action a_t was done as "prediction by mode": given the currently winning unit, $\mathbf{s}^*(\mathbf{u}_t)$, the most likely next map state is calculated as $\widehat{\mathbf{s}}_{t+1} = \text{argmax}_{\mathbf{s}'}(T_{a_t}(\mathbf{s}', \mathbf{s}^*(\mathbf{u}_t))$, resulting in $\widehat{\mathbf{u}}_{t+1} = \mathbf{w}(\widehat{\mathbf{s}}_{t+1})$. In order to avoid

deadlocks, prediction of the currently winning state is suppressed. A sequence of environmental states given a start state $\mathbf{u}_0$ and an action sequence is predicted by consecutively applying one-step predictions on the basis of estimated environmental states; that is, first state $\hat{\mathbf{u}}_1$ is predicted from the start state $\mathbf{u}_0$ and a_0, and each next environmental state $\hat{\mathbf{u}}_{t+1}$ on the basis of $\hat{\mathbf{u}}_t$ and a_t.

3.3.2. Goals and Action Planning. Sequence predictions can be used to plan action sequences in order to reach a goal. This requires the definition of what a goal is in the present context. We suggest defining a goal as a target state or a subset of target states in map space. Target states might be provided by stimulation (e.g., by demonstrating a target environmental state to the system) or might be intrinsically generated as described in the previous paragraph. These target states are then imprinted or memorized, while the system tries to reach and maintain them by executing a suitable sequence of actions. A biological correlate of target state memorization might be persistent nondistractible neural activity found in prefrontal cortex [55]. The described procedure is closely related to the following concepts: (i) one-shot imitation: imprinting the target state corresponds to the single demonstration of the goal, reaching the goal is then done by inference over the learned intuitive physical model. (ii) Active inference: a system predicts a target state and minimizes prediction error by driving the environment towards the predicted (i.e., desired) state.

A large body of reinforcement learning literature exists on how to find a policy $p(a) = \pi(\mathbf{s})$, which specifies how actions should be planned in order to maximize reward. Here we suggest an action planning strategy which does not rely on external reward signals but operates entirely on the distances between target states and the current state. Actually, the drive to try and reach an imprinted goal representation by active inference must in some respect be generated by an intrinsic reward mechanism, which is, however, not explicitly modelled here. Possible distance measures include (here Euclidean) distance either between environmental states, $\|\mathbf{u}^{\text{target}} - \mathbf{u}_t\|$, which is the input prediction error in active inference terminology, or—because of topographic order—between map states, $\|\mathbf{s}^{\text{target}} - \mathbf{s}_t\|$, or both. We found action plans on the basis of environmental state distances to work better than map distance for the tasks considered here; hence, the results in the Method and Results section are generated using this distance measure.

Many distance-based action planning schemes can be imagined; here, we use simple τ step greedy forward search for action planning: on the basis of the true present state $\mathbf{u}_0$, find the action sequence of length τ, the execution of which minimizes the distance between the predicted state resulting from that action sequence and the target state.

4. Method and Results

The system was implemented in *Python*, version 3.7.6 64 bit, using PyTorch's tensor library version 1.4.0, https://pytorch.org/. All experiments have been carried out on a desktop PC equipped with an Intel® Xeon® E-2186G CPU (6 cores, 3.80 GHz) and an NVIDIA Quadro P1000 graphics board. SapSom was tested on the Open AI gym cartpole environment (for a screenshot of the rendered cartpole, see Figure 1(b), inset), https://gym.openai.com/envs/CartPole-v0/. Hence, no external data sets were used, the only data were the variables returned by the gym environment in response to the agent's actions. The SapSom source code has been open-sourced under https://github.com/martinstetter/sapsom.

For the experiments shown, a 16×16 SOM was trained and analyzed as specified below. The environment accepts two actions: push the cart to the left (a_1) or to the right (a_2) with a fixed force and emits four sensory signals, namely, the cart location x, the pole's angle with the vertical θ and their temporal changes (i.e., $\mathbf{u} = (x, \dot{x}, \theta, \dot{\theta})$). The system was originally designed as a testbed for reinforcement learning systems with the goal to keep the pole vertical by balancing; therefore, the environment also returns a reward for each step and triggers a "done" signal; as soon as the pole hits ±15 degrees, the cart hits the screen border, or 200 steps of balancing are successfully executed. Throughout this work, the reward signal was ignored, because SapSom operates in a completely unsupervised way. A sequence of steps between cartpole initialization and trigger of the done signal is referred to as an episode.

The system was trained as follows: in order to assure correct unfolding of the map, the SOM representing the input space was pretrained over 1000 episodes under random action sequences using a standard learning scheme for self-organizing maps with exponential decays for σ and η between start and end values $(8, 0.1)$ and $(0.3, 0.01)$, respectively. Subsequently, both representation and prediction parts were trained simultaneously over 3000 episodes with $\sigma_0 = 4, \eta = \gamma = 0.05$ as follows: for a given state first the SOMs were updated under adaptive neighbourhood, then an action was selected and executed, the resulting next state was observed, and the corresponding transition matrix was updated as specified in the model section. Randomly selected actions were used during "playful exploration."

4.1. Intuitive Physics. Here we tested whether SapSom could learn a representation of the environment's Newtonian dynamics, metaphorically referred to as "intuitive physics" [2]. In technical terms, we tested whether, after training, the system could approximate the real phase portrait of the environment by its own predicted phase portrait. Results are shown for the $\theta - \dot{\theta}$ phase plane because the pole's behaviour rather than the cart's behaviour is usually considered in the cartpole environment.

After training, the real and predicted dynamics of the environment were analyzed by playing real episodes and determining the real and predicted directions of motion from one step to the next. For a given phase point $\mathbf{u}_t$ and next action a_t, the real direction of motion was determined by executing a_t on the environment and calculating $\mathbf{u}_{t+1} - \mathbf{u}_t$. The predicted direction of motion was calculated by applying $\mathbf{u}_t$ and determining $\mathbf{p}_t$, then predicting the next

state $\hat{\mathbf{s}}_{t+1}$ and corresponding predicted input $\hat{\mathbf{u}}_{t+1}$, and finally computing $\hat{\mathbf{u}}_{t+1} - \mathbf{u}_t$.

The real (blue) and predicted (red) directions of motion in the angle phase plane are shown in Figure 2(a) for five complete random episodes. Note that this phase portrait is not uniquely defined, because at each point directions are conditioned on a, x, and $\dot{x}$. Real and predicted directions of motion agree very well with each other. However, there are small deviations, although in principle the cartpole physics is deterministic and should in principle be learnable to arbitrary accuracy. The existence of small deviations is due to the state representation's quantization error: similar, but different environmental state trajectories will be mapped to the same map unit, but will have slightly different time evolutions. These differences cannot be resolved by the system. Where quantization errors become large (e.g., for novel states), prediction errors can become large as well.

Figure 2(b) displays SapSom's predicted directions of motion when planning to execute a left push (blue) or a right push (red), respectively, for the same five episodes. The configuration reflects correct "comprehension" of the situation: a left push generally accelerates the pole to the right; that is, angular velocity increases, which is correctly mirrored by the blue arrows pointing upwards towards increasing $\dot{\theta}$. Under opposite sign, the same is true for right push.

We conclude that SapSom can learn a reasonably accurate representation of the cartpole dynamics only from interventional exploration. Because this is achieved in a completely unsupervised way and without making explicit use of the equations of motion, this can be understood as a way of capturing an intuition about the physics of the environment.

In the next two sections, we present results about inference on this model. In order to separate slow learning and inference effects, learning was switched off in the following by setting $\eta = \gamma = 0$.

4.2. Playing Virtual Episodes. Next, we examined whether the trained system could virtually play episodes on the basis of its intuitive physics; that is, whether given an action sequence and a start state, the corresponding future state sequence could be predicted reasonably well.

Results for a number of different scenarios are summarized in Figure 3. Figure 3(a) illustrates by a number of screenshots how the cartpole evolves under its true dynamics (top row) and under predicted dynamics for the same start state and action sequence (bottom row). Actions were eight left pushes. The bottom row images were generated by manually setting the cartpole's state to $\hat{\mathbf{u}}_t$, rendering the environment, and then capturing the screen. From visual inspection, one may conclude that both sequences agree very well.

A slightly more quantitative analysis of prediction quality is given by plotting the time evolution of x (in arbitrary units as provided by the gym environment) and θ under true (dashed) and predicted (solid) dynamics. Figures 3(b) and 3(c) show how x and θ evolve over time, where blue traces correspond to eight left pushes and red traces to eight right pushes. Figure 3(d) displays the time evolution of θ under five different random action sequences and initial conditions of the cartpole. Corresponding real and simulated dynamics are plotted in the same colour. Although the coincidence is not perfect, the general features of the resulting motions (shifting and tilting to the correct direction) are captured quite well. In order to quantify the prediction quality, simulated and true dynamics were generated repeatedly under 100 different random action sequences and cartpole initializations, and the RMSE between real and predicted angle was computed per time step of prediction horizon. The results are summarized in Table 1. The values moderately increase from 0.028 rad to 0.0423 rad as the number of time steps into the future increases, corresponding to a relative error of 15 to 20 percent when comparing to the absolute range of θ values.

Finally, in order to visualize the prediction performance on a longer and more complicated sequence, the comparison was run under an action sequence of length 39, which elicits an oscillation (Figure 3(e)). The gym environment initializes the cartpole's start state with small random numbers, resulting in a small positive initial value for θ in this case. As a consequence, in the true dynamics (dashed line), a slowly accelerating tilt to the right, in the direction of increasing θ, under gravity is superimposed with the faster oscillation evoked by the action sequence.

The general behaviour (increasing θ under oscillation) is correctly predicted by the agent (solid line), even though the difference between the absolute values of real and predicted angles increases over time as indicated also in Table 1. This increasing prediction error can be understood by keeping in mind that, besides the action sequence, only the start state is available to the system (i.e., no intermediate sync).

It may be concluded that, for the environment considered, SapSom can perform qualitatively and semiquantitatively correct multistep predictions of the environment's future under a given virtual action sequence (usually generated by the agent itself). This encourages us to test the system's action planning performance when performing a task. Since in the present system controlling the environment in order to achieve a goal is an inferential rather than slow learning process, we test SapSom on a set of one-shot imitation tasks.

4.3. One-Shot Imitation. A task requires a goal to be formulated, either implicitly (by reward structure) or explicitly, by demonstrating one or a few success stories, i.e., examples where the goal has been reached. Here we adopt the latter approach, which has the advantage that no sometimes complex reward structure needs to be formulated. For SapSom, we define a goal as a target state or set of target states in its state representation, which it has to reach and maintain. This target state can be spontaneously generated by the system (intrinsic or curiosity-driven goal) or can be imprinted from outside (extrinsic goal).

One-shot imitation refers to the ability of a system "to learn from very few demonstrations of any given task, and

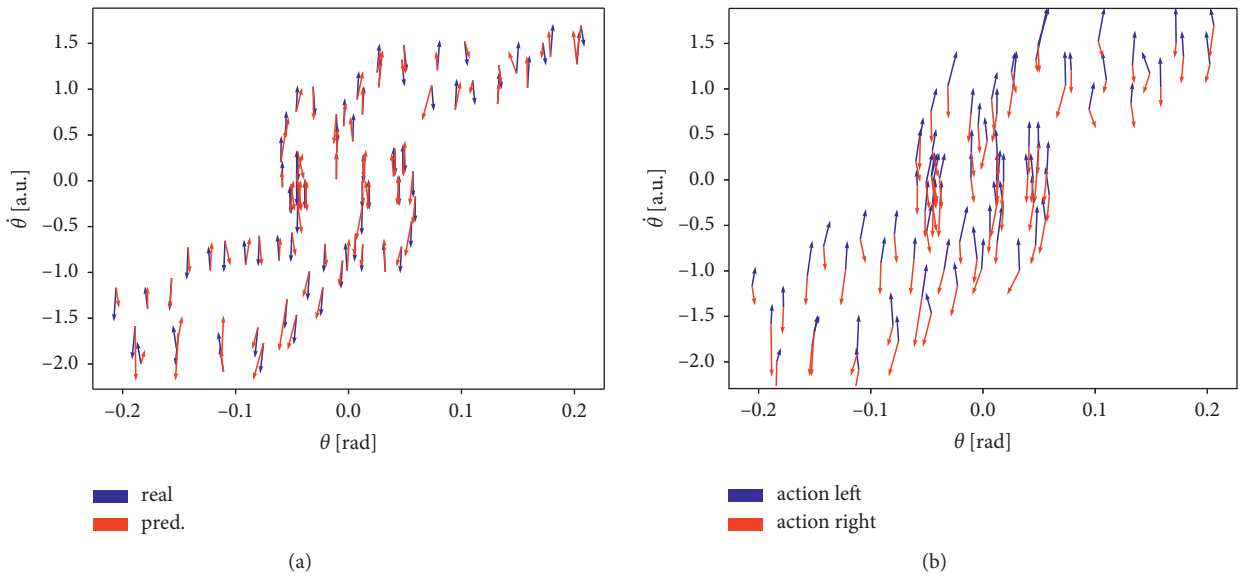

Figure 2: Directions of motion in the $\theta - \dot{\theta}$ phase plane (arrows) for five complete random episodes. Arrows are located at the states $(\theta, \dot{\theta})$ to which they apply. (a) Blue: real directions of motion in the next step as provided by the environment. Red: directions of motion predicted by the network when provided with the same state and action. Predictions approximate real movements very well. (b) Prediction of motions in phase space under virtual left push (blue) and virtual right push (red), respectively (for discussion see text).

instantly generalize to new situations of the same task, without requiring task-specific engineering" [13]. Here we formulate an extrinsic goal by presenting to the system a single sequence of target states, which are imprinted into its map. Imprinting means that the corresponding winning units are memorized as part of the goal. For example, if the goal is to balance the pole, a sequence of states $\mathbf{u}_t = (x_t, \dot{x}_t, 0, 0), t = 0, 1, \ldots, T$ with upright stationary pole under various cart positions and velocities is presented to SapSom. Technically, instead of the true sequence of states, we only present the vector of expected values, $\mathbf{u}^g = E_t[\mathbf{u}_t]$, and the vector of inverse variances or *precisions*, π^g, to the system (the superscript "g" stands for "goal"). For the pole balancing example, the set of goal values for x and $\dot{x}$ will both show a large variance with zero mean. In contrast, the goal values of θ and $\dot{\theta}$ are zero, resulting in zero mean and zero variance for these two variables. As the precision is defined as the inverse variance, in this example the precisions of x and $\dot{x}$ will be very small, and the precisions of θ and $\dot{\theta}$ will be very high. We approximate the low precision values by zero and cap very high precision values to a maximum of 1, resulting in a mean vector of $\mathbf{u}^g = (0, 0, 0, 0)$ and a precision vector of $\pi^g \approx (0, 0, 1, 1)$ for this example.

Reaching the target state then means to search for a sequence of actions, which drive the environment's actual state towards the target state(s) and keep it there. For simplicity, we avoid explicitly determining the distance between the actual state and all target states, but instead use the precision-weighted distance between $\mathbf{u}$ and $\mathbf{u}^g$: $d(\mathbf{u}, \mathbf{u}^g) = \sum_i \pi_i^g (u_i - u_i^g)^2$. For action planning, one-step greedy forward search is applied (i.e., $\tau = 1$).

In this section, we consider three different types of tasks, namely, (i) the (pole) balancing task, (ii) the controlled-tilt task, and (iii) the tilted-balancing task, which are explained in the following. The balancing task consists of keeping the pole upright as stationary and as long as possible. As pointed out above, presenting the goal of a balanced pole results in imprinting $\mathbf{u}^g = (0, 0, 0, 0), \pi^g = (0, 0, 1, 1)$. The controlled-tilt task consists of deliberately letting the pole tilt to a specified side, until the maximum tilt of $\theta^g = \pm 0.2$ rad tolerated by the gym environment is reached. The challenge to the agent is to control the tilt process such that a specific angular velocity $\dot{\theta}^g$ is achieved at the instant where the border is hit. This task is assigned to the agent by imprinting $\mathbf{u}^g = (0, 0, \theta^g, \dot{\theta}^g), \pi^g = (0, 0, 1, 1)$. In the tilted-balancing task, the goal is to keep the pole stationary, but in a tilted position $\theta^g \neq 0$. This task, which is also not easy to achieve by humans, involves letting the pole tilt a little bit at the beginning, followed by a controlled acceleration of the cart to the corresponding side in order to keep the tilt stationary afterwards. This task is assigned to the agent by imprinting $\mathbf{u}^g = (0, 0, \theta^g, 0), \pi^g = (0, 0, 1, 1)$.

Figure 4 shows the time evolution of cartpole location x (Figures 4(a) and 4(c)) and angle θ (Figures 4(b) and 4(d)) for four specific goals given to the agent. The blue traces in Figures 4(a) and 4(b) show 5 attempts of the agent to solve the balancing task. The red traces were produced under a controlled-tilt task with moderate goal angular velocity, $\mathbf{u}^g = (0, 0, 0.2, 0.5)$. The green traces arose under another controlled-tilt task with high negative goal angular velocity $\mathbf{u}^g = (0, 0, -0.2, -5)$.

The final angular velocities were $(0.57, 0.35, 0.40, 0.32, 0.33)$ for the moderate tilt case and $(-2.57, -2.53, -2.46, -2.41, -2.58)$ for the rapid tilt case. The latter angular velocities are about the maximum which can be achieved when pushing to one side all the time, which was correctly predicted by the agent: all but one action over the five rapid tilt trials were "push right."

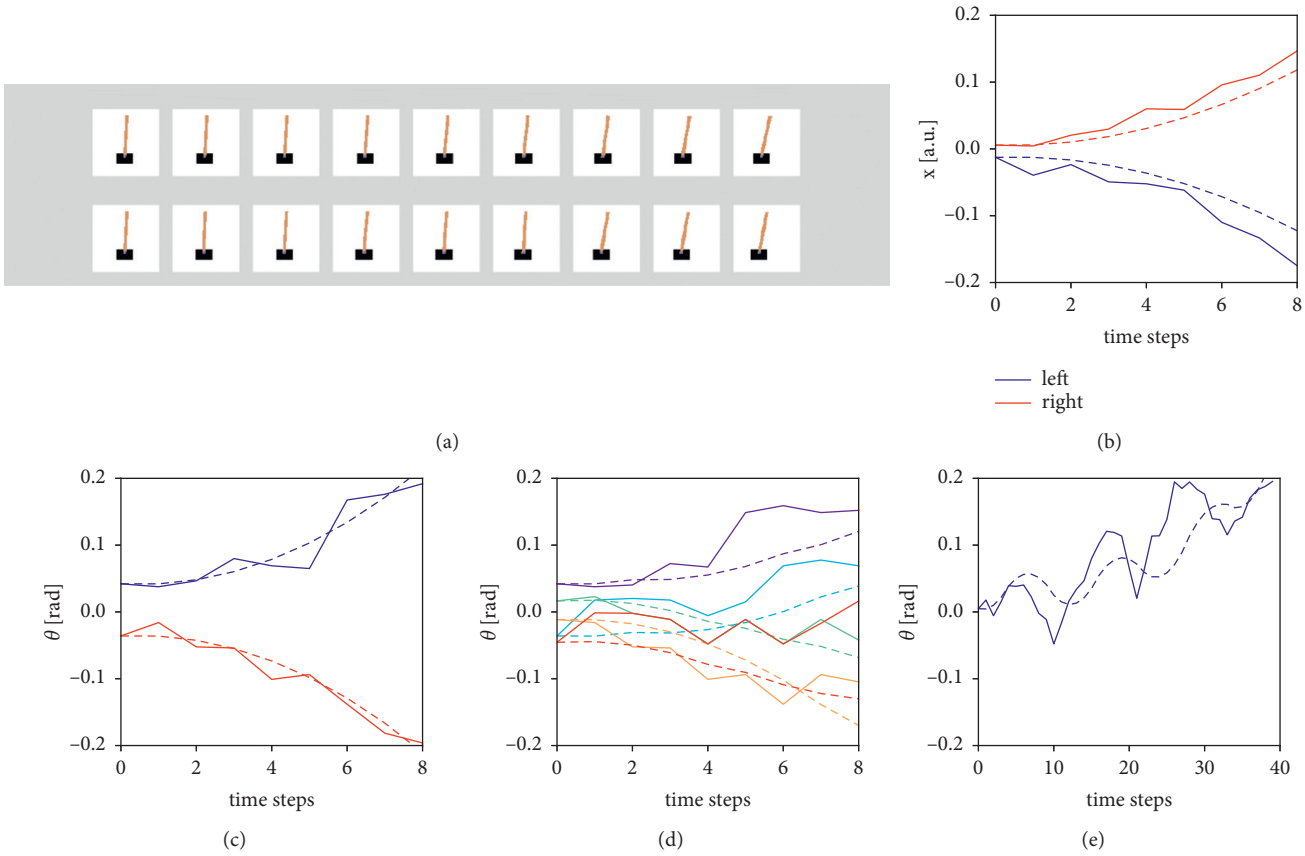

Figure 3: (a) Screenshots of cartpole with identical start state followed by eight left pushes. Top: real time evolution. Bottom: 8-step prediction of time evolution. (b–e) Real (dashed) and predicted (solid) time evolutions of θ and x starting from identical initial states. (b) Time evolution of x and (c) time evolution of θ for the same simulation run. Blue: 8 left pushes; red: 8 right pushes. (d) Time evolution of θ for 5 different random action sequences and cartpole initializations. Traces with same colors correspond to the same simulation run. (e) Time evolution of θ for a longer sequence of length 39: oscillatory actions (3 left followed by alternating (6 x right) (6 x left) pushes). Major features of motion are correctly captured in all cases.

Table 1: RMSE of predicted angle over time steps into the future.

t	1	2	3	4	5	6	7
RMSE	0.0280	0.0293	0.0315	0.0337	0.0396	0.0414	0.0423

Figures 4(c) and 4(d) illustrate how the system acts in a tilted-balancing task with goal angle $\theta^g = 0.15$ rad. To solve this task, the agent had to constantly accelerate the cart into the direction of the tilt in a controlled way such that the cart leaves the regime previously explored and learned. The traces indicate that the agent manages to keep the pole steadily tilted over a considerable number of time steps, although it does not quite reach the goal of $\theta^g = 0.15$ rad but instead stabilizes values around $\theta = 0.08$ rad. Also, when approaching the screen boundaries, the agent fails to stabilize any longer, because this configuration is far from what it experienced during playful exploration with random actions only.

From visual inspection of these examples, we find that the system is quite flexible in solving different related tasks and generalizes satisfactorily to previously unseen regimes. In the following, we quantitatively analyze the performance and variability thereof for a larger number of specific goals for these tasks.

4.3.1. Performance Analysis for the Balancing Task. OpenAI defines "solving" the cartpole problem as being able to balance the pole over an average of 195 time steps per episode, taken over the last 100 episodes, where each episode ends in case of failure or after 200 time steps otherwise. In order to characterize the performance of the agent, we analyzed 100 episodes under the balancing task (Figure 4(b), blue). We found that 61 episodes reached the limit of 200 steps of balancing; the average number of time steps was 188 steps. Hence, SapSom does not quite reach the definition for solving the task but comes quite close.

4.3.2. Performance Analysis for the Controlled-Tilt Task. While keeping the goal angle constant at $\theta^g = 0.2$ rad, we systematically increased the goal angular velocity $\dot{\theta}$ from 0 to 5 in steps of 0.25. For each goal, 20 simulations with randomly initialized cartpoles were run and the mean and standard deviation of the actual angular velocities at the border (i.e., immediately after the done signal of the cartpole

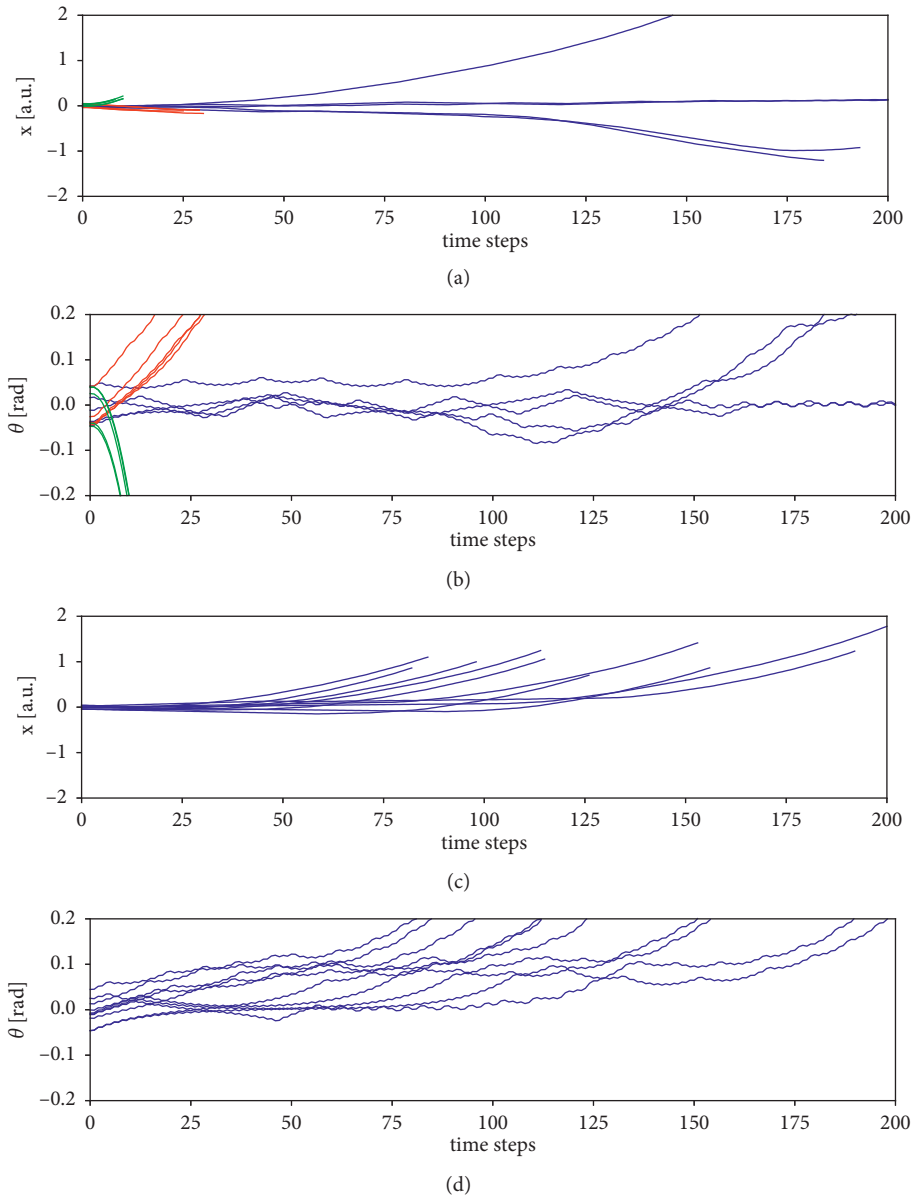

FIGURE 4: Time evolution of x (a, c) and θ (b, d) under the balancing task, two specific controlled-tilt tasks, and a tilted-balancing task. (a, b) Five exemplary traces per task. blue: balancing task; red: controlled tilt to the right with $\dot{\theta}^g = 0.5$; and green: fast controlled tilt to the left with $\dot{\theta}^g = -5$. (c, d) 10 exemplary traces for the tilted-balancing task with $\theta^g = 0.15$ rad.

environment) were computed. First of all, we observed that in every trial the correct angle was approximated; that is, the agent always managed to tilt the pole to the right. The blue trace in Figure 5(a) plots the mean actual angular velocities and their standard deviations (vertical lines). In the range between 0.25 and 1.5, the goal angular velocity could be approximated very closely, whereas the performance dropped for higher goal values. To further analyze this situation, we plotted also the mean excess of the number of left push actions versus right push actions (i.e., $(n_{\text{left}} - n_{\text{right}})/(n_{\text{left}} + n_{\text{right}})$) (Figure 5(a), red line), and the mean number of time steps until the done signal was thrown (Figure 5(a), green line, right axis). It can be seen that already at goal values of 1 and higher, the number of time steps until the border is reached is very small, namely, about 10, and therefore there is only a limited number of possible counts of left versus right actions. In fact, the steps in the actual final angular velocities correspond to steps in the action excess roughly at 0.4 (7 left vs. 3 right pushes), 0.6 (8 left vs. 2 right pushes), and 0.8 (9 left vs. 1 right push). Finally, the goal of zero angular velocity is slightly exceeded; however, from the large number of steps survived, it can be seen that the agent attempts to slowly approximate the border, and the negative action excess reflects its attempt to decelerate the tilt.

4.3.3. Performance Analysis for the Tilted-Balancing Task. While setting the goal angular velocity to zero, we systematically varied the goal angle from −0.2 to 0.2 in steps of 0.025 and ran the task 20 times for each goal. For each run,

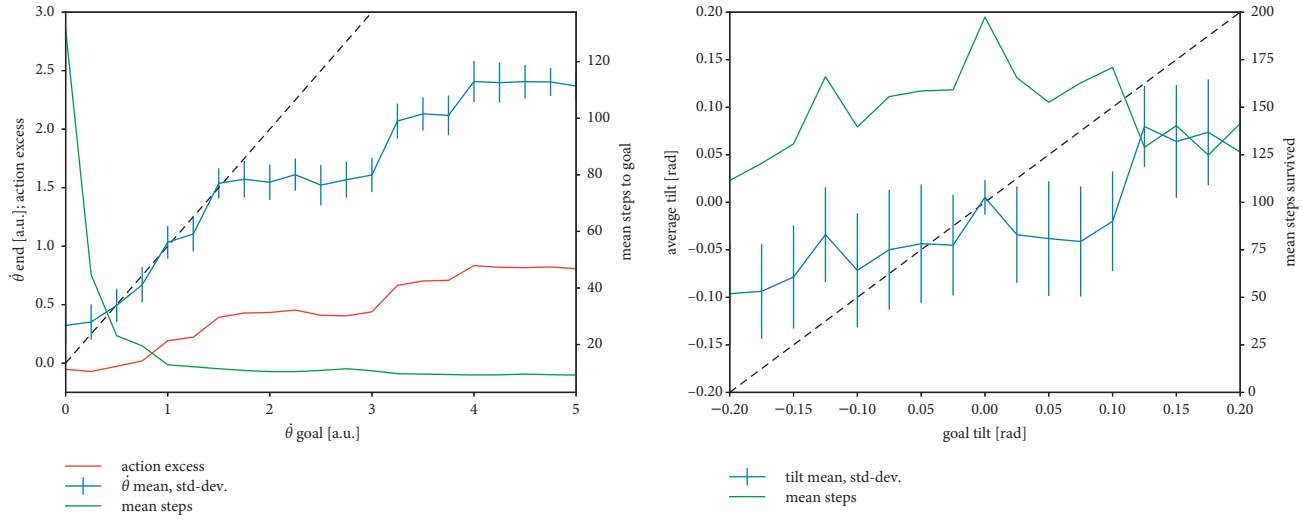

FIGURE 5: (a) Performance analysis for the controlled-tilt task. Blue trace: means and standard deviations (vertical lines) over 20 runs of the actual final angular velocity as function of the goal angular velocity. Black dashed: bisector line. Red (left axis): mean action excess. Green (right axis): mean number of time steps until the done signal. (b) Performance analysis for the tilted-balancing task. Blue trace: means and standard deviations (vertical lines) over 20 runs of the actual average tilt as function of the goal tilt. The actual average tilt was calculated as the mean angle over time steps 50–100. Green (right axis): mean number of steps until the done signal.

we computed the average tilt of the pole over time steps 50–100. This gave the agent some time to build up the tilt initially and excluded the later phase where many trials failed (cf also Figures 4(c) and 4(d)). Where the run survived less than 100 steps, the average was taken between 50 and the end of the run. The blue trace in Figure 5(b) plots the means and standard deviations of the average tilts over 20 runs as a function of the goal tilt (pole angle). While the absolute actual tilts systematically fall below the absolute goal tilts, there is a clear positive trend: the agent clearly attempts tilted balancing and on average succeeds in doing so. The larger standard deviations compared to Figure 5(a) reflect the higher difficulty of the tilted-balancing task as compared to controlled-tilt tasks. The green line (right axis) again displays the mean number of steps survived. For all, even the strong tilts, the agent manages to balance the pole for more than 100 steps on average. The highest survival time is achieved for zero tilt, which corresponds to the simple balancing task.

The results demonstrate that SapSom can perform each of these tasks very well (although not perfectly) after only one presentation of the goal state. This is possible, because task solution is done via inference over the intuitive physics learned, rather than via slow modification of weights. Hence, in comparison with reinforcement learning, the presented system provides two major advantages: (i) it does not need any extrinsic reward structure (which has often to be engineered in a tedious process), and (ii) it can flexibly solve various tasks, which would require both a new reward structure and retraining in reinforcement learning. The latter flexibility has also been specified as an important feature of humanlike learning and performance [2].

5. Discussion

The goal of this work was to suggest a minimal model that shows important properties of artificial intelligence: learning from experience, comprehension of its surroundings, reasoning, planning, and flexible solution of different tasks. It does so by learning a representation of the dynamical physical properties of its environment by exploration and by performing inference on this representation to reach goals. The philosophy behind this approach is related to the KISS principle and Occam's razor at the model level: we identified minimal ingredients which seem both plausible and important for achieving the mentioned properties (there may be completely different ways, though, to generate the same behaviour). The key ingredients of SapSom are as follows:

(i) An adaptive and sparse sensorimotor representation, in particular the possibility to act on an environment rather than just observing, in order to learn by exploration.

(ii) A temporal sequence learning and prediction mechanism at the sensorimotor level.

(iii) A short-term memory mechanism to store target states in state representation space.

(iv) A mechanism to intrinsically generate action and state representations and to reason on their temporal evolution on the basis of temporal sequence prediction.

(v) A distance measure between states. If reasoning is to be done at the representation level, a distance measure between sparse state representations, such as a topographic map, is required.

In the following, we first compare our model to a standard Q-learning setup for pole balancing, then discuss possible extensions of our model, and relate its mechanisms to biological findings.

5.1. Comparison to State-of-the-Art Methods. The proposed agent learns to achieve various different goals on the basis of a state-action and state-transition models that are learned in a completely unsupervised manner. Goals are imprinted by simple activation of states' latent space, for example, as a result of a single presentation of a target environmental state. Hence, it is very easy to switch forth and back between different goals on a per-episode basis for SapSom. In contrast, almost all state-of-the-art methods are related either to explicit physical modelling or to reinforcement learning (RL) in some sense (see Related Work section). While physical simulation engines require the human expert to design this engine, RL-methods, which might be most closely related to our approach, heavily rely on adequate reward signals provided by the environment, which have to be manually designed by the human expert as well. Moreover, the characteristics of the reward signal provided by the environment usually encode or encourage one specific goal only. An RL agent can only learn a new, different task, if (a) the experimenter handcrafts a new reward structure provided to the agent, and (b) the agent undergoes retraining in order to learn to exploit the new reward structure. For that reason, the high flexibility and generalization ability of SapSom alone renders our approach superior to physical engines and RL agents for the context addressed.

However, it is possible to compare the performance of SapSom to that of an RL agent for a stationary task with defined reward structure. Therefore, we reimplemented standard tabular Q-learning [56] for the cartpole environment following frequently used parameter settings. To approximate the number of $16 \times 16 = 256$ states of the SapSom agent, the continuous 4D state space $(x, \dot{x}, \theta, \dot{\theta})$ was discretized to $3 \times 3 \times 6 \times 6 = 324$ bins for the Q-table, giving more emphasis to the important variables θ and $\dot{\theta}$. A learning rate of $\alpha = 0.1$ and an ε-greedy policy with $\varepsilon = 1 - \log_{10}(n/25)$ constrained to $\varepsilon \in [0.1, 1]$ for episode number $n \geq 1$ was used. As SapSom, the agent was trained over 4000 episodes. Of the last 1000 episodes, 87 percent reached the limit of 200 steps of balancing; the average number of time steps was 186. When comparing this to 61 percent completed episodes and 188 average steps of SapSom (see Method and Results section), we find that SapSom reaches a performance comparable to Q-learning, but without manual binning of the input space, without taking any benefit of the reward signal, and while still being flexible enough to immediately perform a different task from the very next episode on. It can be concluded that our model is clearly more effective and efficient than Q-learning and probably related RL methods on the cartpole task.

5.2. Possible Extensions. A simple set of one-layer network architectures was used (in the full model, Figure 1(a), a three-layer architecture) to learn an embedding of the input space. Both the brain and state-of-the art representation techniques, in contrast, maintain hierarchical representations of modalities. In our model, single-layer state representation learning can easily be replaced by hierarchical self-organizing map structures [51], or by contemporary high-performance approaches such as (vector-quantized) variational autoencoders [7, 8], generative adversarial networks [6], or deep convolutional architectures in general (e.g., [3]), which might be trained on raw frame sequences. A sparse representation at the embedding level, which is considered crucial for temporal sequence learning, can be either directly provided by such systems [8], or can be achieved by using a SOM or other competitive learning mechanism [50] on top of their output or encoding layer.

SapSom's temporal sequence learning mechanism is simply 1st-order Markov, rendering it somewhat similar to Hidden Markov Models [57]. Clearly, reasoning on environments of natural complexity requires variable- (and sometimes very high-) order Markov representations. Hawkins et al. [58] developed a biologically plausible model for variable order sequence memory, which is embedded in their hierarchical temporal memory framework. It is based on lateral cortical connections operating on a sparse representation, where each state is represented by multiple model neurons. Their approach could be most naturally incorporated in our framework, but also other approaches for variable-order sequence prediction can be considered without changing the fundamental way the model works [10, 59]. A different obvious possibility to include sensitivity to variable-length history is to use recurrent connectivity. Recurrent SOMs [51, 60] can be trained to represent short sequences of input patterns instead of individual states.

Besides being one-step forward only, action sequence planning is done simply by minimizing the precision-weighted Euclidean distances between current and goal environmental states. For high-dimensional state spaces, however, it is known that the Euclidean distance can lose much of its meaning [61]. Consequently, for high-dimensional spaces, it might be beneficial to either resort to other distance measures, such as, for example, the cosine distance, or to put more weight on distance measurement in the topographically ordered latent space.

Another issue that is unresolved is how to stably establish a hierarchy of temporal timescales which is clearly present in human reasoning. When related to biology, assuming a "temporal timestep" within the brain of a few tens of milliseconds, all reasoning modelled here would operate in the sub second range. Humans, in contrast, make hierarchical action plans over a broad range from below seconds to years and seem to compose longer-term plans by abstracting from shorter ones [18]. Hierarchical arrangements of RSOMs have the potential to model a temporal hierarchy, however, for longer history the algorithm requires some parameter finetuning and becomes subject to numerical instability. Temporal hierarchy might instead be more stably generated when interlacing RSOM layers with a novelty-driven gating mechanism (Stetter, unpublished results).

In its minimal version, SapSom operates in completely unsupervised mode. It does not evaluate any supervisory or extrinsic reward signals. The feedback used is sensory information evoked by its own actions. Because there needs to be some intrinsic motivation mechanism which drives the system to explore its environment, to try and reach goals, and eventually to intrinsically set goals, the present approach shows similarities with curiosity-driven learning, where intrinsic reward signals are generated based on prediction errors [62]. Humans, in contrast, do use both extrinsic and intrinsically generated reward signals (e.g., [63]).

Reinforcement learning mechanisms can be incorporated in a natural way on top of the sensorimotor representation, as SapSom—up to the reward signal—considers its environment as a Markov decision process which it learns to approximate. The present approach therefore shows a natural link with model-based RL [64]. Alternatively, Q-learning could use its update rule to optimize a Q-value that is assigned to each *sa* unit in the sensorimotor map (Figure 1). The resulting policy would then replace the greedy forward search algorithm established here. Generally, however, finding efficient search strategies in action sequence space that approach human performance in being creative and solving problems is an important ongoing research issue.

5.3. Relation to Biology. Conceptually, our model is placed in the middle of the spectrum ranging from approaches that deliberately abstract from the way the brain works [65] to computational neuroscience models which put their main focus on how cognitive mechanisms are specifically implemented in biological neural networks [66]. Our approach is to formulate computational models, which are inspired by fundamental mechanisms of brain function and are in principle compliance with what is known about biological information processing. Accordingly, a few parallels can be drawn between SapSom's present implementation and the brain's biological neural networks.

First of all, Kohonen's self-organizing maps are biologically motivated by retinotopy and smooth feature maps found in the early visual pathway, by Mexican-hat like lateral cortical connectivity and Hebbian learning. These aspects seem more closely related to biology than an error backpropagation mechanism, for which no biological correlate could be found so far. Moreover, SOMs have been used very successfully as models of self-organization in the early visual pathway [67]. Given the observation that all areas in the neocortex are laid out very uniformly and seem to perform similar operations [68], hierarchical systems of SOMs appear to be good candidates of neurally inspired models at least of posterior neocortex.

The state transition matrices optimized during prediction learning can be interpreted as lateral (Figure 1(b)) or top-down (Figure 1(a)) cortico-cortical connections. When doing so, the learning rule (5) just corresponds to spike-time dependent plasticity (e.g., [69]). For this we recall that that map state probabilities $p(\mathbf{s})$ are derived from normalized activations of state neurons (equation (4)). When $T_a(\mathbf{s}', \mathbf{s})$ is interpreted as connection from state-action neuron with index $\mathbf{s}, a$ to target neuron $\mathbf{s}'$, the learning rule reads $\Delta T_a(\mathbf{s}', \mathbf{s}) = (p_{t+1}(\mathbf{s}') - \widehat{p}_{t+1}(\mathbf{s}')) \cdot p_t(\mathbf{s}, a)$. The first term says that T_a will be increased, when $\mathbf{s}'$ becomes active after $\mathbf{s}, a$ (stronger than expected), and is decreased, if $\mathbf{s}'$ is expected active but remains silent.

Moreover, there is evidence that prefrontal cortex is capable of actively maintaining information by robust, persistent neural activity, which, according to prominent models, might be rapidly updatable by a striatum-driven gating mechanism [70]. The system comprised of dorsolateral, anterior cingulate, and orbitofrontal cortices, which interact with the striatum and thalamus, are considered crucial for representing goals and context, generating action plans, and evaluating expected rewards thereof (for a review of data and detailed computational models, see [71]). Hence, nondistractible sustained activation might be a neural correlate of the model's goal states, whereas their distractible counterparts might underly working memory required during mental execution of the search through action plans.

Learning more about the principles of this orchestration process of cortical states, or, in SapSom terminology, learning about how search and prediction should be ideally designed given an implicit world model, might lead to a better understanding of the principles of human thinking in general—an exciting field for future research.

Data Availability

The SapSom source code is available at https://github.com/martinstetter/sapsom.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this study.

Acknowledgments

The authors wish to thank Monika Stetter for numerous valuable discussions on the subject. MS was on sabbatical leave joining the CIML Lab at the University of Regensburg.

References

[1] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.

[2] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and Brain Sciences*, vol. 40, p. e253, 2017.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet: classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105, Curran Associates, Inc., New York, NY, USA, 2012.

[4] A. Graves, D. Eck, N. Beringer, and J. Schmidhuber, "Biologically plausible speech recognition with LSTM neural nets," in *Biologically Inspired Approaches to Advanced Information Technology*, A. J. Ijspeert, M. Murata, and N. Wakamiya, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 127–136, 2004.

[5] A. Graves, Ar Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," 2013.

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, and S. Ozair, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 2672–2680, Cambridge, MA, USA, December 2014.

[7] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013.

[8] A. Razavi, A. Van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," 2019.

[9] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[10] D. Rawlinson, A. Ahmed, and G. Kowadlo, "Learning distant cause and effect using only local and immediate credit assignment," 2019.

[11] J. Pearl, "Theoretical impediments to machine learning with seven sparks from the causal revolution," 2018, https://arxiv.org/abs/1801.04016.

[12] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, "Generalizing from a few examples: a survey on few-shot learning," 2019, https://arxiv.org/abs/1904.05046.

[13] Y. Duan, M. Andrychowicz, B. Stadie, J. Ho, J. Schneider, and I. Sutskever, "One-shot imitation learning," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, and S. Vishwanathan, Eds., pp. 1087–1098, Curran Associates, Inc., New York, NY, USA, 2017.

[14] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, and R. Munos, "Learning to reinforcement learn," 2016, https://arxiv.org/abs/1611.05763.

[15] J. X. Wang, Z. Kurth-Nelson, D. Kumaran et al., "Prefrontal cortex as a meta-reinforcement learning system," *Nature Neuroscience*, vol. 21, no. 6, pp. 860–868, 2018.

[16] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[17] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[18] S. S. Khemlani, R. Mackiewicz, M. Bucciarelli, and P. N. Johnson-Laird, "Kinematic mental simulations in abduction and deduction," *Proceedings of the National Academy of Sciences*, vol. 110, no. 42, pp. 16766–16771, 2013.

[19] K. Friston, "Learning and inference in the brain," *Neural Networks*, vol. 16, no. 9, pp. 1325–1352, 2003.

[20] K. Friston, "A theory of cortical responses," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 360, no. 1456, pp. 815–836, 2005.

[21] K. Friston, "The free-energy principle: a unified brain theory?" *Nature Reviews Neuroscience*, vol. 11, no. 2, pp. 127–138, 2010.

[22] R. P. N. Rao and D. H. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature Neuroscience*, vol. 2, no. 1, pp. 79–87, 1999.

[23] J. Hohwy, *The Predictive Mind*, Oxford University Press, Oxford, UK, 2013.

[24] A. Clark, *Surfing Uncertainty: Prediction, Action, and the Embodied Mind*, Oxford University Press, Oxford, UK, 2016.

[25] M. Stetter and E. W. Lang, "Learning intuitive physics and one-shot imitation using state-action-prediction self-organizing maps," 2021, https://arxiv.org/abs/2007.01647.

[26] E. S. Spelke and K. D. Kinzler, "Core knowledge," *Developmental Science*, vol. 10, no. 1, pp. 89–96, 2007.

[27] J. Schmidhuber, "Artificial scientists & artists based on the formal theory of creativity. artificial general intelligence," in *Proceedings of the third conference on artificial general intelligence*, Lugano, Switzerland, March 2010.

[28] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum, "Simulation as an engine of physical scene understanding," *Proceedings of the National Academy of Sciences*, vol. 110, no. 45, pp. 18327–18332, 2013.

[29] M. Toussaint, "A sensorimotor map: modulating lateral interactions for anticipation and planning," *Neural Computation*, vol. 18, no. 5, pp. 1132–1155, 2006.

[30] P. Morasso, V. Sanguineti, and G. Spada, "A computational theory of targeting movements based on force fields and topology representing networks," *Neurocomputing*, vol. 15, no. 3-4, pp. 411–434, 1997.

[31] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum, "Galileo: perceiving physical object properties by integrating a physics engine with deep learning," *Advances in Neural Information Processing Systems*, vol. 28, pp. 127–135, 2015.

[32] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, and k Kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., pp. 4502–4510, Curran Associates, Inc., New York, NY, USA, 2016.

[33] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti, "Visual interaction networks: learning a physics simulator from video," *Advances in Neural Information Processing Systems*, vol. 30, pp. 4539–4547, 2017.

[34] M. B. Chang, T. Ullman, A. Torralba, and J. B. Tenenbaum, "A compositional object-based approach to learning physical dynamics," 2017, https://arxiv.org/abs/1612.00341.

[35] S. van Steenkiste, M. Chang, K. Greff, and J. Schmidhuber, "Relational neural expectation maximization: unsupervised discovery of objects and their interactions," 2018, https://arxiv.org/abs/1802.10353.

[36] K. Fragkiadaki, J. Huang, A. Alemi, S. Vijayanarasimhan, S. Ricco, and R. Sukthankar, "Motion prediction under multimodality with conditional stochastic networks," 2017, https://arxiv.org/abs/1705.02082.

[37] R. Riochet, M. Castro, M. Bernard, A. Lerer, R. Fergus, and V. Izard, "IntPhys: a framework and benchmark for visual intuitive physics reasoning," 2018, https://arxiv.org/abs/1803.07616.

[38] D. Zheng, V. Luo, J. Wu, and J. B. Tenenbaum, "Unsupervised learning of latent physical properties using perception-prediction networks," 2018, https://arxiv.org/abs/1807.09244.

[39] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, and M. Malinowski, "Relational inductive biases, deep learning, and graph networks," 2018, https://arxiv.org/abs/1806.01261.

[40] J. Wu, E. Lu, P. Kohli, W. T. Freeman, and J. B. Tenenbaum, "Learning to see physics via visual de-animation," *Advances in Neural Information Processing Systems*, vol. 30, pp. 153–164, 2017.

[41] S. Ehrhardt, A. Monszpart, N. J. Mitra, and A. Vedaldi, "Learning a physical long-term predictor," 2017, https://arxiv.org/abs/1703.00247.

[42] K. Smith, L. Mei, S. Yao, J. Wu, E. Spelke, and J. Tenenbaum, "Modeling expectation violation in intuitive physics with coarse probabilistic object representations," in *Advances in Neural Information Processing Systems*, H. Wallach,

H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds., vol. 32pp. 8985–8995, 2019.

[43] K. Kansky, T. Silver, D. A. Mély, M. Eldawy, M. Lázaro-Gredilla, and X. Lou, "Schema networks: zero-shot transfer with a generative causal model of intuitive physics," 2017, https://arxiv.org/abs/1706.04317.

[44] S. Ehrhardt, A. Monszpart, N. Mitra, and A. Vedaldi, "Unsupervised intuitive physics from visual observations," 2019, https://arxiv.org/abs/1805.05086.

[45] S. Ehrhardt, A. Monszpart, N. J. Mitra, and A. Vedaldi, "Taking visual motion prediction to new heightfields," *Computer Vision and Image Understanding*, vol. 181, pp. 14–25, 2019.

[46] T. Wu, J. Peurifoy, I. L. Chuang, and M. Tegmark, "Meta-learning autoencoders for few-shot prediction," 2018, https://arxiv.org/abs/1807.09912.

[47] A. Ajay, M. Bauza, J. Wu, N. Fazeli, J. B. Tenenbaum, and A. Rodriguez, "Combining physical simulators and object-based networks for control," 2019, https://arxiv.org/abs/1904.06580.

[48] H. Nguyen, J. Patravali, F. Li, and A. Fern, "Learning intuitive physics by explaining surprise," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1539–1542, Seattle, WA, USA., June 2020.

[49] T. Ye, X. Wang, J. Davidson, and A. Gupta, "Interpretable intuitive physics model," in *Lecture Notes in Computer Science*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Springer, Berlin, Germany, pp. 89–105, 2018.

[50] P. Földiák, "Forming sparse representations by local anti-Hebbian learning," *Biological Cybernetics*, vol. 64, no. 2, pp. 165–170, 1990.

[51] J. W. Miller and P. H Lommel, "Biomimetic sensory abstraction using hierarchical quilted self-organizing maps of society of photo-optical instrumentation engineers (SPIE) Conference series," 2006.

[52] D. Rawlinson and G. Kowadlo, "Generating adaptive behaviour within a memory-prediction framework," *PLoS ONE*, vol. 7, no. 1, Article ID e29264, 2012.

[53] J. Pearl, *Causality: Models, Reasoning, and Inference*, Cambridge University Press, Cambridge, UK, 2000.

[54] M. Stetter, "Dynamic functional tuning of nonlinear cortical networks," *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 73, Article ID 031903, 2006.

[55] J. M. Fuster and G. E. Alexander, "Neuron activity related to short-term memory," *Science*, vol. 173, no. 3997, pp. 652–654, 1971.

[56] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, UK, 2nd edition, 2018.

[57] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, pp. 1554–1563, 1966.

[58] J. Hawkins, D. George, and J. Niemasik, "Sequence memory for prediction, inference and behaviour," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 364, no. 1521, pp. 1203–1209, 2009.

[59] S. Jensen, D. Boley, M. Gini, and P. Schrater, "Rapid on-line temporal sequence prediction by an adaptive agent," *Proceedings of the International Conference on Autonomous Agents*, vol. 23, pp. 67–73, 2005.

[60] M. Varsta, J. Heikkonen, and J. Millan, "Context learning with the self-organizing map," in *Proceedings of the Workshop on Self-Organizing Maps*, pp. 197–202, Espoo, Finland, June 1997.

[61] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, *On The Surprising Behavior of Distance Metrics in High Dimensional Space*, Springer, Berlin, Heidelberg, 2001.

[62] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proceedings of the 34th International Conference on Machine Learning. ICML 2017. JMLR.org*, pp. 2778–2787, New York, NY, USA, August 2017.

[63] W. Schultz, P. Dayan, and P. R. Montague, "A neural substrate of prediction and reward," *Science*, vol. 275, no. 5306, pp. 1593–1599, 1997.

[64] M. M. Botvinick and A. Weinstein, "Model-based hierarchical reinforcement learning and human action control," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, 2014.

[65] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall Press, USA, 3rd edition, 2009.

[66] G. Deco and E. T. Rolls, "Attention, short-term memory, and action selection: a unifying theory," *Progress in Neurobiology*, vol. 76, no. 4, pp. 236–256, 2005.

[67] K. Obermayer, H. Ritter, and K. Schulten, "A principle for the formation of the spatial structure of cortical feature maps," *Proceedings of the National Academy of Sciences*, vol. 87, no. 21, pp. 8345–8349, 1990.

[68] V. B. Mountcastle, "An organizing principle for cerebral function: the unit module and the distributed system," in *Neuroscience, Fourth Study Program*, F. O. Schmitt, Ed., pp. 21–42, MIT Press, Cambridge, MA, USA, 1979.

[69] G. Bi and Mm Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type," *Journal of Neuroscience*, vol. 18, pp. 10464–10472, 1999.

[70] R. C. O'Reilly, "Biologically based computational models of high-level cognition," *Science*, vol. 314, no. 5796, pp. 91–94, 2006.

[71] R. C. O'Reilly, J. Russin, and S. A. Herd, "Computational models of motivated frontal function," in *The Frontal Lobes of Handbook of Clinical Neurology*, M. D'Esposito and J. H. Grafman, Eds., pp. 317–332, Elsevier, Amsterdam, Netherlands, 2019.